AD-767 710

OPTIMAL ALLOCATIONS IN THE CONSTRUCTION OF k-OUT-OF-n RELIABILITY SYSTEMS

Sheldon M. Ross, et al

California University

Prepared for:

Office of Naval Research
Army Research Office-Durham

September 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

ORC 73-17
SEPTEMBER 1973

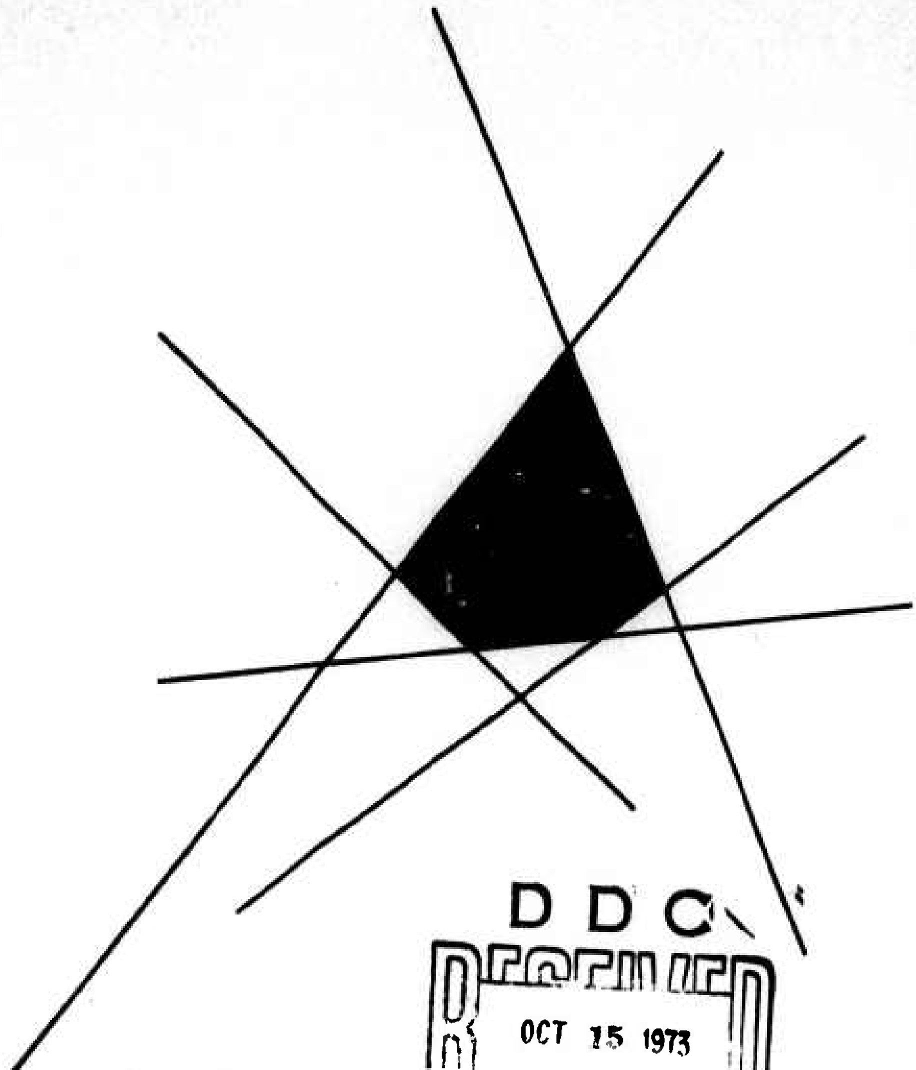
OPTIMAL ALLOCATIONS IN THE CONSTRUCTION OF k-OUT-OF-n RELIABILITY SYSTEMS

by

SHELDON M. ROSS,

CYRUS DERMAN and

GERALD J. LIEBERMAN



OPERATIONS
RESEARCH
CENTER

26

COLLEGE OF ENGINEERING
UNIVERSITY OF CALIFORNIA • BERKELEY

OPTIMAL ALLOCATIONS IN THE CONSTRUCTION OF k-OUT-OF-n RELIABILITY SYSTEMS

by

Sheldon M. Ross†
Department of Industrial Engineering
and Operations Research
University of California, Berkeley

and

Cyrus Derman
Civil Engineering
Division of Mathematical Methods
and Operations Research
Columbia University
New York, New York

and

Gerald J. Lieberman
Operations Research Department
Stanford University
Stanford, California

SEPTEMBER 1973 ORC 73-17

†This research has been partially supported by the U.S. Army Research Office-Durham under Contract DA-31-124-ARO-D-331 and the Office of Naval Research under Contract N00014-69-A-0200-1036 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of California, Berkeley	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

OPTIMAL ALLOCATIONS IN THE CONSTRUCTION OF k-OUT-OF-n RELIABILITY SYSTEMS

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Research Report

5. AUTHOR(S) (First name, middle initial, last name)

Sheldon M. Ross, Cyrus Derman and Gerald J. Lieberman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1973	21	1

8a. CONTRACT OR GRANT NO. N00014-69-A-0200-1036 b. PROJECT NO. NR 042 238 c. Research Project No.: WW 041 d.	9a. ORIGINATOR'S REPORT NUMBER(S) ORC 73-17 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES Also supported by the U.S. Army Research Office-Durham under Contract DA-31-124-ARO-D-331.	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Department of the Navy Arlington, Virginia 22217

13. ABSTRACT

SEE ABSTRACT.

ia

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Reliability						
k-Out-of-n						
Optimal Policy						
Sequential Case						
Allocating Resources						
Dynamic Programming						

ABSTRACT

We want to build n components so as to form an n component system which will function if at least k of the components function. If x dollars is invested in building a component, then this component will function with probability $P(x)$. Given a total income of A dollars, the problem of interest is to determine how much money we should invest in each component so as to maximize the probability of attaining a functioning system. This problem is considered both in the sequential and in the nonsequential case. Conditions under which it is optimal to allocate A/n units at each stage, when A is your initial fortune, are presented. The special case $P(x) = \min(x,1)$ is also considered in detail.

OPTIMAL ALLOCATIONS IN THE CONSTRUCTION OF
k-OUT-OF-n RELIABILITY SYSTEMS

C. Derman, G. J. Lieberman, S. M. Ross

1. Introduction.

We want to build n components so as to form an n component system which will function if at least k of the components function. If x dollars is invested in building a component then this component will function with probability $P(x)$, where $P(x)$ is an increasing function such that $P(0) = 0$. We have a total income of A dollars. The problem of interest is to determine how much money we should invest in each component so as to maximize the probability of attaining a functioning system. We will be interested in this problem both in the sequential and in the nonsequential case. In the sequential case we assume that the individual components are built sequentially in time and that knowledge as to whether or not a component functions is available to us before we have to allocate our investment in the next component. In the nonsequential case it is assumed that all allocations must be simultaneously made.

In Section 2 of this paper we consider the case $k = 1$ and present conditions on $P(x)$ under which it is optimal to put an equal investment in all n components and conditions under which it is optimal to put the total fortune A into a single component. In Section 3, it

is shown that the "equal investment" condition carries over to the case of general k. In Section 4 we consider the special case $P(x) = x$ in the sequential situation and determine the optimal policy when $k = 2$. A conjecture as to the optimal policy in the general case is also made. Several remarks are made concerning the non-sequential case with $P(x) = x$ (considered in [1]) in Section 5. In the final section we consider a related problem.

2. The Case $k = 1$.

When $k = 1$ the sequential and nonsequential cases are identical; both are involved in determining $\underline{x} = (x_1, \dots, x_n)$ with $x_i \geq 0$, $\sum_1^n x_i = A$ so as to maximize $[1 - \prod_1^n (1 - P(x_i))]$, the reliability in the latter case, and the identical expression $p(x_1) + \sum_{i=2}^{n} \prod_{j=1}^{i-1} (1-p(x_j))\, p(x_i)$ in the former case. The interpretation of $\underline{x}$ is that x_i dollars is to be invested in component i in the nonsequential case, and, in the sequential case, x_i dollars is to be invested in the ith attempt if the first $i-1$ attempts to build a functioning component are unsuccessful. The above is equivalent to choosing $x_1, \dots, x_n$, $x_i \geq 0$, $\sum_1^n x_i = A$ so as to minimize $\sum_{i=1}^{n} \log(1 - P(x_i))$.

Proposition 1:

(a) If $\log(1 - P(x))$ is convex then the optimal allocation is

$$x_1 = x_2 = \cdots = x_n = A/n$$

(b) If $\log(1 - P(x))$ is concave then an optimal allocation is

$$x_1 = A,\ x_2 = \cdots = x_n = 0 \ .$$

Proof: Follows from standard results about concave and convex functions. ∎

Remarks:

(i) In part (b) the condition that $\log(1 - P(x))$ be concave can be weakened to the condition that $\log(1 - P(x))$ be subadditive, i.e., that is, $(1 - P(x + y)) \leq (1 - P(x))(1 - P(y))$.

(ii) The condition that $\log(1 - P(x))$ be superadditive, i.e., $(1 - P(x + y)) \geq (1 - P(x))\ (1 - P(y))$, would not be sufficient to establish part (a). It would, however, necessarily imply that the optimal $\underline{x}$ vector would have all positive components.

3. The General Case.

Part (a) of Proposition 1 remains true in the general case.

Theorem 1: If $\log(1 - P(x))$ is (strictly) convex then when one wants to sequentially build k working components in at most n attempts, $n \geq k$, then it is (uniquely) optimal to allocate A/n at each stage when A is your total resources.

Proof: Assume first that $\log(1 - P(x))$ is strictly convex. The proof is by induction on k, and, as we have already proven the result when $k = 1$, let us assume that it is true for all values less than k. Now consider the k component case. If the number of possible stages is k then a policy reduces to a vector $(x_1, \ldots, x_k)$ where $x_i \geq 0$, $\sum_1^k x_i \leq A$, with the interpretation that the policy invests x_1 in the first stage and if the first i-1 stages all result in working components then x_i is invested in the ith stage. (That is, when the number of available stages is identical with the number of desired components then the sequential problem reduces to the nonsequential one.) Thus, the problem reduces to

$$\text{Maximizing } \prod_{i=1}^{k} P(x_i) \text{ subject to } x_i \geq 0, \sum_1^k x_i = A$$

or, equivalently, to,

$$\text{Maximizing } \sum_1^k \log P(x_i) \text{ subject to } x_i \geq 0, \sum_1^k x_i = A.$$

Now the strict convexity of $\log(1 - P(x))$ implies that $\log(P(x))$ is strictly concave (see the following Lemma 1). Hence, by standard arguments, it follows that $x_i = A/k$, $i = 1, \ldots, k$ is the (unique) optimal allocation and the result is established in this case.

Thus the result is true in the k component case when the number of stages is also equal to k. So let us assume that the result holds in the k component case whenever the number of stages is less than n and try to prove that it also holds when the number of stages

equals n. To do so suppose that an optimal policy† for a k-component n-stage problem initially allocates an amount x_1. Now if this initial attempt is successful then the induction hypothesis (on the number of desired components) tells us that it is uniquely optimal from that point on to allocate $(A-x_1)/(n-1)$ for each of the remaining $n-1$ stages. On the other hand even if the initial attempt is unsuccessful then as there are only $n-1$ stages to go it follows by the induction hypothesis on the number of stages in the k-component case that it is still optimal to allocate $(A-x_1)/(n-1)$ on each of the remaining $n-1$ stages. If $x_1 = A/n$ then the result is proven. So let us suppose that $x_1 \neq A/n$ and obtain a contradiction. If $x_1 \neq A/n$ then consider the policy that allocates $\frac{1}{2}[x_1 + (A-x_1)/(n-1)]$ for each of the first two stages and then allocates $(A-x_1)/(n-1)$ for each of the last $n-2$ stages. This policy is thus identical with the optimal policy during the last $n-2$ stages. Now the probability that at least one of the components built during the first two stages is successful is

$$1 - [1 - P(x_1)]\left[1 - P\left(\frac{A-x_1}{n-1}\right)\right] \tag{1}$$

under the optimal policy; while it is

$$1 - \left[1 - P\left(\frac{1}{2}\left(x_1 + \frac{A-x_1}{n-1}\right)\right)\right]^2 \tag{2}$$

†It is easy to show that an optimal policy exists for the n stage problem. This is done by first proving recursively that the optimal value function for an n stage problem is a continuous function of the initial fortune whenever $P(x)$ is continuous. An optimal policy then exists since a continuous function obtains its maximum on a closed set.

under the new policy. Also the probability that both components are successful is

$$(3) \qquad P(x_1)\, P\left(\frac{A-x_1}{n-1}\right)$$

for the optimal policy; while it is

$$(4) \qquad P^2\left(\frac{1}{2}\left[x_1 + \frac{A-x_1}{n-1}\right]\right)$$

for the new policy. It follows by the results given for $k = 1$ that (2) is greater than (1), and it follows from that fact that $\log(P(x))$ is concave (Lemma 1) that (4) is greater than (3). Hence, under the new policy the number of successes during the first two stages is stochastically greater than it is under the optimal policy. As the two policies are identical after the first two stages it thus follows that the probability of at least k successes is greater under the new policy than it is under the optimal policy. This contradiction shows that $x_1 = A/n$, which proves the result in the k-component case for any number of stages, which also completes the initial induction proof.

If $\log(1 - P(x))$ is convex but not strictly so, then we can approximate $\log(1 - P(x))$ arbitrarily closely by functions that are strictly convex and then apply a continuity argument. ∎

The following lemma was used in the proof of the theorem.

Lemma 1: If $0 \leq P(x) \leq 1$, and $\log(1 - P(x))$ is convex then $\log(P(x))$ is concave.

Proof: Suppose the hypothesis of the lemma are true. Then

$$0 \leq \frac{d^2}{dx^2} \log(1 - P(x)) = \frac{P(x)\, P''(x) - (P'(x))^2 - P''(x)}{(1 - P(x))^2}$$

implying that

$$P''(x) \leq \frac{-(P'(x))^2}{1 - P(x)} \leq 0$$

Now,

$$\frac{d^2}{dx^2} \log P(x) = \frac{P(x)\, P''(x) - (P'(x))^2}{(P(x))^2}$$

which is negative since $P''(x) \leq 0$. ∎

Remarks.

(i) It follows from Theorem 1 that when $\log(1 - P(x))$ is convex the optimal sequential policy is nonsequential is nature, and is thus also the optimal policy when the allocations for each stage must be made simultaneously rather than sequentially.

(ii) If we think of $P(x)$ as being a probability distribution function then the condition that $\log(1 - P(x))$ be convex is equivalent to the condition that $P(x)$ is a decreasing failure rate distribution. Since mixtures of decreasing failure rate distributions are themselves decreasing failure rate distributions it thus follows that $\log(1 - P(x))$ will be convex whenever

$$P(x) = \int_\alpha P_\alpha(x)\, dF(\alpha)$$

where

$$\log(1 - P_\alpha(x)) \text{ is convex for all } \alpha,$$

and

$$F(\alpha) \text{ is a probability distribution function.}$$

In particular, any $P(x)$ of the form

$$P(x) = \int_\alpha (1 - e^{-\alpha x})\, dF(\alpha)$$

will be such that $\log(1 - P(x))$ is convex.

(iii) If we let $V_n(A)$ denote the probability that at least k successes will occur in the n stages under an optimal policy then from Theorem 1 it follows that

$$V_n(A) = \sum_{i=k}^{n} \binom{n}{i} (P(A/n))^i (1 - P(A/n))^{n-i}$$

If $P(x)$ is differentiable then

$$\lim_{x \to 0} \frac{P(x)}{x} = \lim_{x \to 0} P'(x) = P'(0) \equiv \lambda;$$

and thus from the Poisson approximation to the binomial distribution it follows that

$$V_n(A) \uparrow e^{-\lambda A} \sum_{i=k}^{\infty} (\lambda A)^k/k! \qquad \text{as} \quad n \uparrow \infty$$

(This is so since $V_n(A) = \text{Prob}\{\text{Bin}(n, P(A/n)) \geq k$ where $\text{Bin}(n, P(A/n))$ represents a binomial random variable with parameters n and $P(A/n)$.) It should be pointed out that this convergence is not necessarily monotone for an arbitrary differentiable function $P(x)$ with $P(0) = 0$.

4. The Sequential Case $P(x) = x$.

While the analogue of part (a) of Proposition 1 remains true in the case of arbitrary k it is obvious that the same cannot be said of part (b). In this section we consider the special case $P(x) = x$.

Suppose that we need to sequentially build two ($k = 2$) functioning components in at most n attempts when $P(x) = \min(x,1)$. Suppose that our initial fortune is A and consider the policy π which sequentially allocates A/n at each stage until a functioning component is built, and, at this point allocates all of the remaining fortune for the next stage. In other words if our present fortune is y and at most r additional components can be built then

(a) if two additional functioning components are still needed then π allocates y/r for the next component;

(b) if only one additional functioning component is needed then π allocates y for the next component.

Denote by $U_n(y)$, the probability that two functioning components will be built when our initial fortune is y and policy π is employed.

Proposition 2: For $y \leq 1$

$$U_n(y) = (1 - \frac{y}{n})^n + y - 1$$

Proof: By conditioning on the time of the first successful component we see that

$$U_n(y) = \sum_{r=1}^{n-1} (1 - \frac{y}{n})^{r-1} \frac{y}{n} (y - \frac{ry}{n})$$

which simplifies to prove the proposition. ∎

The next proposition states that $U_n(y)$ satisfies what, in dynamic programming terminology, is known as the optimality equation.

<u>Proposition 3</u>: For $y \leq 1$

$$U_n(y) = \underset{0 \leq x \leq y}{\text{Max}} \; [x(y-x) + (1-x) \; U_{n-1}(y-x)]$$

<u>Proof</u>: Define

$$f(x) = x(y-x) + (1-x) \; U_{n-1}(y-x)$$

$$= x(y-x) + (1-x) \left[\left(1 - \frac{y-x}{n-1}\right)^{n-1} + y - x - 1 \right]$$

Differentiation yields

$$f'(x) = (1-x) \left(1 - \frac{y-x}{n-1}\right)^{n-2} - \left(1 - \frac{y-x}{n-1}\right)^{n-1}$$

implying that $f'(x) = 0$ if and only if

$$1 - x = 1 - \frac{y-x}{n-1}$$

or

$$x = \frac{y}{n}$$

Since $f''(y/n) = 0$, it follows that $f(x)$ attains its maximum value at $x = y/n$. This proves the result since $f(y/n) = U_n(y)$. ■

<u>Theorem 2</u>: For an initial fortune of $A \leq 1$ the policy π maximizes the probability of obtaining two functioning components.

Proof: From Proposition 3 we see that

$$(5) \qquad U_n(y) \geq x(y-x) + (1-x)\, U_{n-1}(y-x) \qquad \text{for all} \quad 0 \leq x \leq y$$

The right side of the above can be interpreted as the return (i.e., probability of obtaining two functioning components) if x is allocated for the first stage and then policy π is used for the remaining stages. As the inequality (5) holds for all x, $0 \leq y \leq x$, and, as we already know that π is optimal when only one additional functioning component is needed (Proposition 1), we can interpret (5) as stating that using policy π is better than doing anything else for one stage and then switching to π. Repeating this argument yields that using π is better than doing anything else for two stages and then switching to π. Finally, by repeating the same argument a total of n times we see that using policy π is better than doing anything else for the first n stages. This completes the proof. ∎

Remark:

Since Theorem 2 states that $U_n(y)$ is the maximal probability when at most n stages are available it follows that $U_n(y)$ is an increasing function of n and thus, for $0 \leq y \leq 1$,

$$U_n(y) = \left(1 - \frac{y}{n}\right)^n + y - 1 \uparrow e^{-y} + y - 1 \qquad \text{as} \quad n \uparrow \infty$$

(This constitutes another proof that the convergence of $\left(1 - \frac{y}{n}\right)^n$ to e^{-y} is monotone when $0 \leq y \leq 1$.) In fact, for any positive y it it well known and can be easily shown that the convergence is monotone for all $n > N$ where $y/N < 1$.

The policy π is no longer optimal when our initial fortune can be greater than 1. Let us define the policy π^* to be such that when the present fortune is y and at most n additional components can be built.

(a) allocates y if only one additional working component is needed; and

(b) if two additional working components are needed, allocates

$$\begin{array}{ll} y - 1 & \text{if } y \geq \frac{n}{n-1} \\ \frac{y}{n} & \text{if } y \leq \frac{n}{n-1} \end{array}$$

If we let $V_n(y)$ denote the probability of a success (two working components) when our initial fortune is y and policy π^* is employed then, as $y \leq n/(n-1)$ implies that $y - \frac{y}{n} \leq \frac{n-1}{n-2}$, it follows that, for $0 < y < 2$

$$V_n(y) = \begin{cases} U_n(y) & \text{if } y \leq \frac{n}{n-1} \\ y - 1 + (2-y)\, U_{n-1}(1) & \text{if } y > \frac{n}{n-1} \end{cases}$$

or, equivalently, for $0 < y < 2$

$$V_n(y) = \begin{cases} (1 - \frac{y}{n})^n + y - 1 & \text{if } y \leq \frac{n}{n-1} \\ (2-y)\, (\frac{n-2}{n-1})^{n-1} + y - 1 & \text{if } y \geq \frac{n}{n-1} \end{cases}$$

<u>Theorem 3</u>: Policy π^* is optimal.

<u>Proof</u>: The proof would follow exactly as in the proof of Theorem 2, if we could show that

(6) $V_n(y) \geq x \operatorname{Min}(y-x,\ 1) + (1-x)\, V_{n-1}(y-x)$ for all $x \leq \operatorname{Min}(1,y)$

When $y \leq 1$, (6) is identical to (5) and thus we only need prove (6) when $1 < y < 2$. Hence we must show that, for $1 < y < 2$,

$$(7)\quad V_n(y) \geq \operatorname{Max}\Big[\max_{0 \leq x \leq y-1} \{x + (1-x)\, V_{n-1}(y-x)\},\ \max_{y-1 \leq x \leq 1} \{x(y-x) + (1-x)\, V_{n-1}(y-x)\}\Big]$$

We consider 2 cases.

<u>Case 1</u>. $y \leq \frac{n}{n-1}$.

Now,

$$\max_{0 \leq x \leq y-1} \{x + (1-x)\, V_{n-1}(y-x)\}$$

$$= \max_{0 \leq x \leq y-1} \{x + (1-x)\, [(1 - \tfrac{y-x}{n-1})^{n-1} + y - x - 1]\}$$

Define the function $f(x)$ by

$$f(x) \equiv x + (1-x)\, [(1 - \tfrac{y-x}{n-1})^{n-1} + y - x - 1]$$

Now,

$$f'(x) = 1 + (1-x)[(1 - \tfrac{y-x}{n-1})^{n-2} - 1] - (1 - \tfrac{y-x}{n-1})^{n-1} - y + x + 1$$

and

$$f''(x) = (1-x)\, (\tfrac{n-2}{n-1})\, (1 - \tfrac{y-x}{n-1})^{n-3} - (1 - \tfrac{y-x}{n-1})^{n-2} + 2 - (1 - \tfrac{y-x}{n-1})^{n-2} \geq 0$$

Hence $f(x)$ is a convex function in the region $0 \leq x \leq y-1$, and thus obtains its maximum value in this region either at $x = 0$ or $x = y-1$. Now

$$f(0) = \left(1 - \frac{y}{n-1}\right)^{n-1} + y - 1$$

and

$$f(y-1) = y - 1 + (2-y)\left(\frac{n-2}{n-1}\right)^{n-1}$$

Now the function

$$g(y) = f(y-1) - V_n(y) = (2-y)\left(\frac{n-2}{n-1}\right)^{n-1} - \left(1 - \frac{y}{n}\right)^n$$

is zero at $y = \frac{n}{n-1}$. Differentiation shows that it is an increasing function when $y \leq \frac{n}{n-1}$ and thus it follows that

$$f(y-1) \leq y - 1 + \left(1 - \frac{y}{n}\right)^n = V_n(y)$$

As was previously noted $\left(1 - \frac{y}{n}\right)^n$ is increasing in n, it follows that $f(0) \leq V_n(y)$; thus,

$$\max_{0 \leq x \leq y-1} f(x) \leq V_n(y) \quad \text{when} \quad y \leq \frac{n}{n-1}$$

Also, since when $y \leq \frac{n}{n-1}$, the inequality

$$V_n(y) \geq \max_{y-1 \leq x \leq 1} \{x(y-x) + (1-x)\, V_{n-1}(y-x)\}$$

is identical to the inequality (5), it follows that (7) is established when $y \leq \frac{n}{n-1}$. We are thus ready for

<u>Case 2</u>: $y \geq \frac{n}{n-1}$

In this case

$$\underset{y-1 \leq x \leq 1}{\text{Max}} \{x(y-x) + (1-x)\, V_{n-1}(y-x)\}$$

$$= \underset{y-1 \leq x \leq 1}{\text{Max}} \{x(y-x) + (1-x)\, [(1 - \frac{y-x}{n-1})^{n-1} + y - x - 1]\}$$

Now the function in brackets above is a decreasing function of x for $x \geq y-1$ (to show this we use the fact that $x \geq y-1$, $y \geq \frac{n}{n-1}$ implies that $nx \geq y$), and thus the Max above is equal to

$$= y - 1 + (2-y)\, (\frac{n-2}{n-1})^{n-1}$$

$$= V_n(y)$$

Hence, if we can show that $V_n(y) \geq \underset{0 \leq x \leq y-1}{\text{Max}} g(x)$ when $y \geq \frac{n}{n-1}$, where $g(x)$ is defined by

$$g(x) = x + (1-x)\, V_{n-1}(y-x) ,$$

then the proof will be complete.

Now, let us suppose that $y \geq \frac{n-1}{n-2}$. Then

$$\underset{0 \leq x \leq y-1}{\text{Max}} g(x) = \text{Max} \left[\underset{0 \leq x \leq y - \frac{n-1}{n-2}}{\text{Max}} g(x), \quad \underset{y - \frac{n-1}{n-2} \leq x \leq y-1}{\text{Max}} g(x) \right]$$

But

$$\underset{0 \leq x \leq y - \frac{n-1}{n-2}}{\text{Max}} g(x) = \underset{0 \leq x \leq y - \frac{n-1}{n-2}}{\text{Max}} \{x + (1-x)[(2-y+x)\, (\frac{n-3}{n-2})^{n-2} + y - x - 1]\}$$

Now the function in brackets above can be easily shown, upon differentiation, to be a convex function and thus it attains its maximum value either at

$x = 0$ or at $x = y - \frac{n-1}{n-2}$. Thus

$$(8) \qquad \max_{0 \le x \le y - \frac{n-1}{n-2}} g(x) = \mathrm{Max}[g(0),\ g(y - \tfrac{n-1}{n-2})]$$

Also,

$$\max_{y - \frac{n-1}{n-2} \le x \le y-1} g(x) = \max_{y - \frac{n-1}{n-2} \le x \le y-1} \{x + (1-x)[(1 - \tfrac{y-x}{n-1})^{n-1} + y-x-1]\}$$

Now it has previously been shown that the above function in brackets is a convex function and thus

$$(9) \qquad \max_{y - \frac{n-1}{n-2} \le x \le y-1} g(x) = \mathrm{Max}[g(y - \tfrac{n-1}{n-2}),\ g(y-1)]$$

Hence, from (8) and (9) we see that when $y \ge \frac{n-1}{n-2}$

$$(10) \qquad \max_{0 \le x \le y-1} g(x) = \mathrm{Max}\begin{bmatrix} (2-y)\left(\frac{n-3}{n-2}\right)^{n-2} + y - 1 \\ y - \frac{n-1}{n-2} + \left[\frac{2n-3}{n-2} - y\right]\left[\left(\frac{n-3}{n-2}\right)^{n-1} + \frac{1}{n-2}\right] \\ y - 1 + (2-y)\left(\frac{n-2}{n-1}\right)^{n-1} \end{bmatrix}$$

On the other hand, when $\frac{n-1}{n-2} > y \ge \frac{n}{n-1}$, we obtain that

$$(11) \qquad \max_{0 \le x \le y-1} g(x) = \mathrm{Max}\begin{bmatrix} (2-y)\left(\frac{n-3}{n-2}\right)^{n-2} + y - 1 \\ (2-y)\left(\frac{n-2}{n-1}\right)^{n-1} + y - 1 \end{bmatrix}$$

With the help of a little algebra we obtain from (10) and (11) that when $y \geq \frac{n}{n-1}$

$$\max_{0 \leq x \leq y-1} g(x) = y - 1 + (2-y)\left(\frac{n-2}{n-1}\right)^{n-1} = V_n(y)$$

and the proof is complete.

Hence the optimal policy is determined in the case $k = 2$. If $n = \infty$, it can be seen that no optimal policy exists. The dynamic programming functional equation will have a solution; however, the solution is not the return function of any policy. The policy determined by the solution is the non-optimal policy of allocating $x = 0$ at every stage.

In the general case consider the policy π^* which is such that if our present fortune is y and if k additional working components are needed with at most n stages to go, then π^* calls for allocating

$$\begin{array}{lll} \frac{y}{n} & \text{if} & y \leq \frac{n}{n-1}(k-1) \\ \\ y - (k-1) & \text{if} & y \geq \frac{n}{n-1}(k-1) \end{array}$$

(Note that π^* corresponds to the previously defined π^* when $k = 2$.) Define $V_{n,k}(y)$ to be the probability of success under π^* when our present fortune is y and k additional working components are needed and at most n additional components can be built. If we can show that

$$(12)\quad V_{n,k}(y) \geq \max_{0 \leq x \leq \min(y,1)} [xV_{n-1,k-1}(y-x) + (1-x)\, V_{n-1,k}(y-x)]$$

then it would follow that π^* is the optimal strategy. We have, at present, been unable to verify (12) but we conjecture that it is valid and that π^* is optimal.

A simple formula for $V_{n,k}(y)$ in the case $y < 1$ is obtained by conditioning on the number of steps required to obtain $k-1$ successful components. This yields

$$V_{n,k}(y) = \sum_{r=k-1}^{n-1} \binom{r-1}{k-2} \left(\frac{y}{n}\right)^{k-1} \left(1 - \frac{y}{n}\right)^{r-k+1} \left(y - \frac{ry}{n}\right), \quad 0 < y < 1.$$

5. <u>The Non-Sequential Case</u>; $P(x) = x$.

The non-sequential case with $P(x) = x$ and general k was considered in [1] where it was shown that the optimal $\underline{x}$ vector $\underline{x}^* = (x_1^*, \ldots, x_n^*)$ is such that all of the non-zero element of $\underline{x}^*$ are equal. A problem of interest is to determine for $0 < y \leq k$ that value of r that maximizes

$$\begin{aligned} Q(r) &= P_r\{\text{at least } k \text{ components work}\} \\ &= \sum_{j=k}^{r} \binom{r}{j} \left(\frac{y}{r}\right)^j \left(1 - \frac{y}{r}\right)^{r-j} \end{aligned}$$

for $r = k, k+1, \ldots, n$. From Proposition 1(b), $r = 1$ when $k = 1$. In general the optimal value of r is a function of y and k but at present only seems obtainable by numerical methods. However, we can make several remarks:

(1) If y is near enough to k then the optimal value of r is $r = k$. To see this note that when $y = k$, $Q(k) = 1$ and $Q(r) < 1$

if $r > k$. Since $Q(r)$ is a continuous function of y for each r the substance of the remark follows.

(2) For every r there exists an $\epsilon_r > 0$ such that for $y < \epsilon_r$, $Q(r+1) \geq Q(r)$ with strict inequality holding if $k > 1$. Thus, loosely speaking, for small y the amount of redundancy in the optimal allocation is large. We show this by considering for $r \geq k$

$$\lim_{y \to 0} \frac{Q(r)}{Q(r+1)} = \left(1 + \frac{1}{r}\right)^k \left(1 - \frac{k}{r+1}\right)$$

$$= R(k) \text{ (say)}.$$

We need only show that $R(k) \leq 1$ with strict inequality holding for $k > 1$. Now $R(1) = 1$, $R(2) = 1 - 1/r^2 < 1$. However, as long as $r \geq k+1$,

$$\frac{R(k+1)}{R(k)} = \frac{\left(\frac{r+1}{r}\right)^{k+1} \left(1 - \frac{k+1}{r+1}\right)}{\left(\frac{r+1}{r}\right)^{k} \left(1 - \frac{k}{r+1}\right)}$$

$$= \frac{r+1}{r} \cdot \frac{1 - \frac{k+1}{r+1}}{1 - \frac{k}{r+1}}$$

$$= \frac{r+1}{r} \cdot \frac{r-k}{r+1-k}$$

$$= \frac{1 - \frac{k}{r}}{1 - \frac{k}{r+1}}$$

$$< 1 .$$

Then, $R(k) < 1$ for all $k > 1$ and the proof of the remark follows.

6. A Related Problem.

Rather than maximizing the probability of sequentially building k successful components within a budget of A we shall now assume that our budget is unlimited and that the problem is to minimize the expected amount of money spent in obtaining the k successful components.

Consider first the case $k = 1$. If x is initially allocated and the component built is not successful (which will occur with probability $1 - P(x)$) then the situation will be exactly the same as it was before the initial investment. Hence, if it was initially optimal to allocate x then it will still be optimal to allocate x for the second component. Hence a policy corresponds to a value x (in dynamic programming terminology we are restricting attention to stationary policies), and the expected cost to obtain a functioning component under such a policy would equal $x/P(x)$. Thus, the problem is to

$$\min_{x > 0} \frac{x}{P(x)}$$

It should perhaps be pointed out that when $\log(1 - P(x))$ is convex then $x/P(x)$ is an increasing function of x and hence no optimal policy exists. Similarly if $\log(1 - P(x))$ is superadditive then

$$\inf_{x > 0} \frac{x}{P(x)} = \lim_{x \to 0} \frac{x}{P(x)} = \frac{1}{P'(0)}$$

and again no optimal policy exists.

The case of general k is solved by noting that it is identical to the $k = 1$ problem taken k times. Hence, an optimal (if one exists) policy would be to invest x^* units at each stage until a total of k working components are obtained, where x^* (if it exists) is such that

$$\frac{x^*}{P(x^*)} = \min_{x > 0} \frac{x}{P(x)} .$$

REFERENCE.

[1] Derman, C., Lieberman, G.J., Ross, S.M., Assembly of Systems Having Maximum Reliability (to appear in ONR Logistic Quarterly).